并行文件系统中 Disk Cache 一致性协议及其性能分析

武北虹
(厦门大学计算机科学系 厦门 361005)

摘 要 本文在并行文件系统中引入 disk cache 多复本技术，从而为并行计算机提供高性能的文件系统. 对于 disk cache 多复本间数据一致性维护，本文提出了"主从式"和"对称式"两类方法，并从其应用的通用性角度，基于等概率模型，对各类方法以及 disk cache 单复本系统进行了性能分析和比较.

关键词 并行计算机，并行文件系统，disk cache，一致性协议.

DISK CACHE COHERENCE PROTOCOLS AND THEIR PERFORMANCE ANALYSIS IN PARALLEL FILE SYSTEMS

Wu Beihong
(*Department of Computer Science, Xiamen University, Xiamen* 361005)

Abstract Multiple block copies of disk cache is introduced into parallel file system in this paper so as to provide high performance file systems for parallel computers. This paper gives several disk cache coherence protocols, and on the basis of the equal probability model, analyses their average performance in contrast with those having single disk cache copy.

Keywords Parallel computer, parallel file system, disk cache, coherence protocol.

1 引 言

随着科学技术的迅猛发展，在气象、石油、理论物理等许多领域中，人们对计算的速度和精度提出了越来越高的要求. 超高速的并行计算机已成为现代科学技术的至高点，在高科技和基础研究中都发挥着极重要的作用. 并行操作系统是并行机研究的一个重要组成部分，在这方面出现了以 MACH[1]，AMOEBA[2]等为代表的一些较成功的并行操作系统. 但它们的文件系统是针对仅由一个处理结点控制所有 I/O 设备的体系结构的传统单一文件系统，这种体系结构使得 I/O 速度与并行机的超高速处理能力之间形成了十分悬殊的差距，严重影响了整个系统的性能.

本文 1994 年 6 月 7 日收到. 本课题得到国家自然科学基金资助. **武北虹**，讲师，获博士学位，从事并行处理方面的研究工作.

为改善上述 I/O 瓶颈现象，出现了 PID 结构，即将 I/O 设备的管理分散到多个 I/O 结点上，进行并行 I/O 操作. 这种体系结构要求对文件系统进行新的研究，引入适合硬件结构的并行文件系统(如 Intel ipsc/2 的 CFS[3]及国内"曙光二号"的 PFS[4]).

一般的文件系统由接口、映射、缓冲和驱动四层构成. 在硬件条件一定的情况下，缓冲区技术是整个文件系统性能的关键. 在单机文件系统中由于采用了这种被称为"disk cache"的内存缓冲区技术而大大提高了文件系统的性能，如 UNIX 中的块设备管理[5]. D. Kotz 等人已证实，并行文件系统中引入 disk cache 技术同样会非常显著地提高系统性能[6].

在以往的并行文件系统研制中，由于主要是为面向科学计算的并行机而配置，它们的文件处理模式绝大多数是对一个大文件的顺序读写，故其 disk cache 技术均采用单复本(如 CFS 及 PFS)，即每个 I/O 结点的 disk cache 中仅有本结点上的盘块映象. 如果采用将逻辑文件存储块交叉地均匀分布在不同结点所控制的磁盘上的负载分配策略，那么当多个计算结点同时完成对一个大文件的顺序读写时，各 I/O 结点只要负责对本结点所带磁盘的盘块进行管理，即可实现最大程度的 I/O 并行[4].

但是科学计算毕竟是计算机应用中的一小部分，从并行机将来发展的通用性角度考虑，其文件处理模式将显示出很大的随机性. 若采用 disk cache 单复本技术，当多个结点同时对同一 I/O 结点控制的盘块进行访问时，又将发生 I/O 瓶颈. 为此引入 disk cache 多复本技术，即各 I/O 结点的 disk cache 中可以有其它结点上的盘块映象. 当然，随之带来的就是这些 disk cache 多复本的一致性维护问题，本文将提出一些维护方法并对其进行性能分析.

2　模型系统体系结构

为研究方便，将各种结构(如 Mesh[4]，Supercube[3]，DASH[9]等) 的消息传递型并行机系统中用于并行文件系统的部分抽象成 n 个 I/O 结点通过互联网络进行通信的结构(见图 1)，以此作为研究对象.

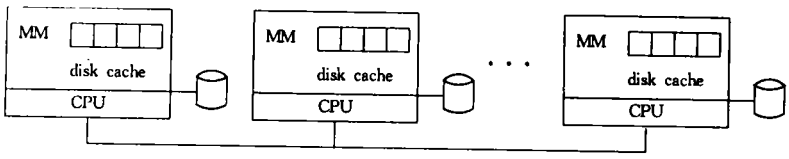


图 1　模型系统体系结构

该并行文件系统在硬件的支持下，将磁盘管理分散到各 I/O 结点上，并将它们作为一个逻辑上的大磁盘来处理，逻辑文件存储块交叉地均匀分布在不同结点控制的磁盘上. 文件组织采用类似于 UNIX 的索引方式，由文件说明块、文件索引块和文件存储块构成. 限于篇幅，本文对此不加详述，直接由它所提供的 mapping 技术来获得一 I/O 请求的文件逻辑块所对应的 I/O 结点以及其上的盘块，重点将研究 disk cache 一层中关于该块的操作. 将一文件的逻辑块 LB 对应的物理盘块所属结点称为 LB 的本地结点；除本地结点外 disk cache 中有 LB 映象的其它结点称为 LB 的复本结点；发出关于 LB 的 I/O 请求的结点称为 LB 的请求结点.

该模型中各 CPU 间进行通信的 send，receive 原语格式为：

send/receive(命令名,源结点号,目的结点号,盘块号,数据的内存起始地址)

各参数分别用cmd,src,dst,bno及buf表示.根据cmd的不同,buf可有可无,dst可以是多个结点,表示同时给多个结点发送同一命令或数据.在 src 和 dst 中分别用 rno,ono,cno 和 acno 表示 LB 的请求结点号、本地结点号、某复本结点号及所有复本结点号,cmd 内容及含义见表 1.

表 1

RDBLK	读一块
WTBYTE	写一字节
RDDATA	发送或接收读出的块数据
WTDATA	向本地结点磁盘回写一块数据
INVALID	通知复本结点使某一 disk cache 块失效或通知本地结点使其它复本结点上 disk cache 块均失效
TODIRTY	通知某结点将某 disk cache 块的状态变为 DIRTY
SWDATA	发送或接收是否需要向磁盘回写某块数据的标志
ELIMINATE	通知本地结点某结点上的某 disk cache 块已失效

3 两类一致性维护协议

关于并行文件系统中 disk cache 一致性问题,国内外很少有参考文献谈及.基于硬件 cache 中一些思想的启发[7—9]并通过对 disk cache 具体特点的分析,本文提出"主从式"和"对称式"两类协议,它们对一文件逻辑块 LB 在各结点中的 disk cache 映象间的地位关系有着不同的处理.

下文对各协议的说明采用带动作的 disk cache 块状态转换图。在这些图中,

$(s_1) \xrightarrow{event(A)} (s_2)$:表示当该 disk cache 块所在结点上发生关于它对应的 LB 的事件 event 时,该结点要完成动作 A 且该 disk cache 块从状态 s_1 变为 s_2.

$(s_1) \xrightarrow{cmd(A)} (s_2)$:表示当该 disk cache 块所在结点收到其它结点发来的关于该块的命令 cmd 时,该结点要完成动作 A 且该 disk cache 块从状态 s_1 变为 s_2.

$\xrightarrow{event(A)} (s) \xrightarrow{cmd(A)} (s)$: 表示 event 事件发生前或收到 cmd 命令前,LB 在该结点上不存在 disk cache 映象,发生或收到后才以状态 s 存在,并完成动作 A.

$\xrightarrow{cmd(A)}$:表示收到 cmd 命令前后,LB 在该结点上均不存在 disk cache 映象,仅指出该结点对 cmd 的处理动作 A.

event 包括"读失效(read miss)"、"写命中(write hit)"、" 写失效(write miss)"和"被淘汰(replace)"四种.淘汰算法首先选择内容无效的块,若没有则采用精确 LRU 算法进行淘汰.

3.1 "主从式"维护方法

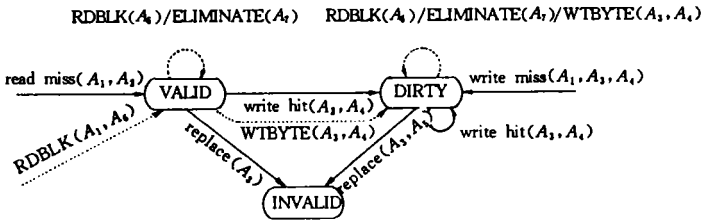
这类方法的特点是以本地结点上的复本为"主",其余复本的存在均依赖于本地结点上的复本.当请求结点失效时,其所需数据均由本地结点提供.若本地结点上的复本被淘汰,则其它结点上的复本不能继续存在.

根据当一结点在执行写操作时对其它复本的不同处理方法,此类方法又分为 WIOC(Write_Invalid_Other_Copies)及 WTAC(Write_Through_All_Copies)两种,前者使其它复本全部失效,而后者将其同时写入所有复本中.这两种方法中 disk cache 块的数据结构是相同的,每块除其数据内容及盘块号外,引入另外三个标志(state,owner,copy):state 用于表示该

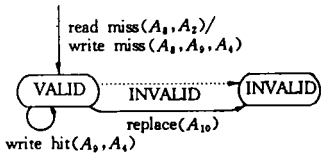
块的当前状态，若 state=INVALID，则该块内容无效，否则该块内容有效，可被使用. 当 state≠INVALID 时，owner 指出该块所在结点是否为其内容对应的文件逻辑块 LB 的本地结点(1 是，0 不是). 若 owner=1，其有效状态有 VALID 和 DIRTY(该块内容已与相应盘块不同)两个；若 owner=0 则仅有 VALID 一个有效状态. copy 为 LB 当前所有复本结点号的集合，仅 state≠INVALID 且 owner=1 的块其 copy 是有效的，LB 的其它 disk cache 映象所在结点均要到其本地结点上来获得其 copy 信息，因为如果每一复本均有一个 copy 标志，则为维护这些 copy 间的一致性又要增加许多开销，使问题复杂化.

“主从式”方法的 WIOC 协议的详细描述见图 2.

WTAC 协议与 WIOC 的不同仅在于发生 write hit 时，A_3 中 send WTBYTE 命令(而不是 INVALID 命令)，且 copy 不变；(b)图中在 VALID 状态时可以接收 WTBYTE 命令，进行动作 A_4 且状态不变.



(a) LB 在其本地结点上 disk cache 映象状态转换图



(b) LB 在其非本地结点上的 disk cache 映象状态转换图

其中：

A_1:{read into disk cache from disk;
owner←1;copy←ϕ}
A_2:{read demanded bytes from disk cache}
A_3:{if (copy-{rno}≠ϕ){
acno←copy-{rno};
send(INVALID, ono, acno, bno);
copy←copy-acno}
}
A_4:{write demanded bytes to disk cache}
A_5:{write back to disk}
A_6:{send(RDDATA, ono, rno, bno, buf);
copy←copy∪{rno}}
A_7:{copy←copy-{rno}}
A_8:{send(RDBLK, rno, ono, bno);
receive (RDDATA, ono, rno, bno, buf);
put into disk cache;
owner←0;
}
A_9:{send(WTBYTE, rno, ono, bno, buf)}
A_{10}:{send(ELIMINATE, rno, ono, bno)}

图 2　“主从式”WIOC 协议：带动作状态转换图

3.2　“对称式”维护方法

与前类方法相比，这类方法的特点体现了各复本间的相对对称性关系. 当请求结点失效时，其所需数据可由任一有此块复本的结点提供. 当本地结点上的复本被淘汰时，其它结点上的复本仍能独立存在.

这类方法在数据结构上每个处理机除与前类方法相同的 disk cache 缓冲池外，增加一个复本标志缓冲池 CFB. CFB 中的每一缓冲区指出某一文件逻辑块 LB 在其本地结点上的 disk cache 映象已不存在，但在其它结点的 disk cache 中有其复本存在. LB 的这种信息只可能保存在其本地结点的 CFB 中. 由于每个 CFB 缓冲区仅由盘块号及 copy 标志(与前面定义同)组成，故 CFB 所占内存空间并不大.

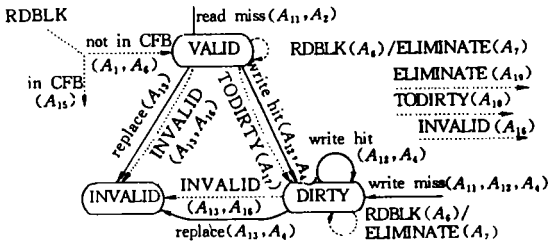
这类方法也有 WIOC 及 WTAC 两种. 任一 disk cache 块均有 INVALID，VALID 及

DIRTY 三种状态. WIOC 及 WTAC 协议见图 3,图 4. 从此也可看出与"主从式"相比所具有的对称性.

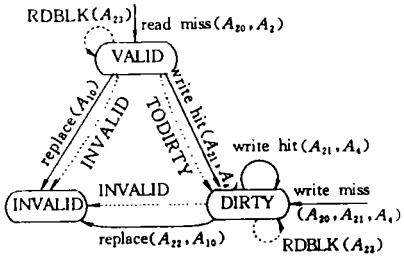
4 性能分析

在本文研究的 disk cache 数据一致性系统中,选用 release 一致性模型[10],假设 send 为双向通信,即要求对方收到命令后发一个回答信息给源结点. 同时硬件能提供操作系统实现同步操作(acquire,release)的硬件同步支持. 可以证明,在这两个条件下,由上述各方法进行维护的 disk cache 系统具有 release 一致性.

本文研究的出发点是从并行机发展的通用性角度来考虑,其文件访问模式呈现出随机性,那么将所有应用领域作为一个整体,从大量应用的统计学观点来看,这种访问符合等概率访问,故本文将基于等概率模型对各方法进行性能分析.



(a) LB 在其本地结点上 disk cache 映象的状态转换图



(b)LB 在其非本地结点上 disk cache 映象的状态转换图

其中:

A_{11}:{if (in CFB){
cno←any node in "copy" of CFB buffer;
send(RDBLK, rno, cno, bno);
receive(RDDATA, cno, rno, bno, buf);
put into disk cache; owner←1;
copy←"copy" of CFB buffer;
free this CFB buffer;}
else A_1}

A_{12}:{if (copy≠ϕ){
acno←copy;copy←ϕ;
send(INVALID, ono, acno, bno);}
}

A_{13}:{if (copy≠ϕ)allocate a buffer in CFB and
put "blkno", "copy" into it }

A_{14}:{if (copy≠ϕ){
cno←any node in "copy";
send(TODIRTY, ono, cno, bno);}
else A_5}

A_{15}:{cno←any node in "copy" of CFB buffer;
" copy" of CFB buffer←"copy" of CFB buffer ∪
{rno};
send(RDBLK, rno, cno, bno);}

A_{16}:{acno←"copy" of CFB buffer-{rno};
if (acno≠ϕ) send(INVALID, ono, acno, bno);
"copy" of CFB buffer←{rno};}

A_{17}:{send(SWDATA, ono, rno, bno, 0)};

A_{18}:{acno←"copy" of CFB buffer-{rno};
if (acno ≠ϕ){
cno←any node in acno;
send(TODIRTY, ono, cno, bno);
send(SWDATA, ono, rno, bno, 0)}
else {
send(SWDATA, ono, rno, bno, 1);
receive(WTDATA, rno, ono, bno, buf);
A_5}
}

A_{19}:{acno←"copy" of CFB buffer-{rno};
if (acno≠ϕ) "copy" of CFB buffer←acno;
else free this CFB buffer}

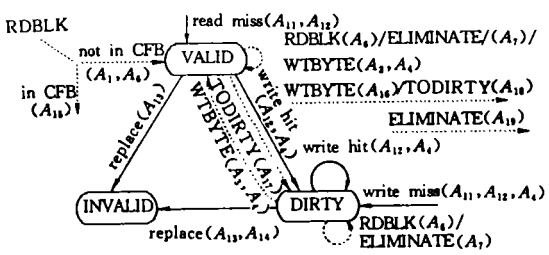
A_{20}:{send(RDBLK, rno, ono, bno);
receive(RDDATA, ono/cno, rno, bno,buf);
put into disk cache; owner←0}

A_{21}:{send(INVALID, rno, ono, bno)}

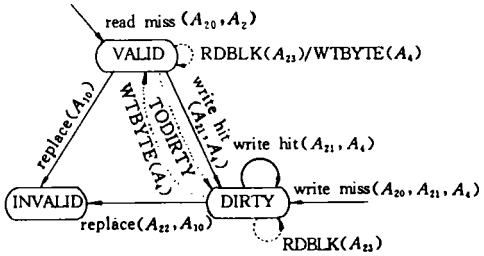
A_{22}:{send(TODIRTY, rno, ono, bno; &answer);
receive(SWDATA, ono, rno, bno, &answer);
if(answer==1) send(WTDATA, rno,
ono, bno, buf);
}

A_{23}:{send(RDDATA, con, rno, bno, buf)}

图 3 对称式 WIOC 协议:带动作的状态转换图



(a) LB 在其本地结点上 disk cache 映象的状态转换图



(b)LB 在其非本地结点上的 disk cache 映象的状态转换图

图 4　对称式 WTAC 协议：带动作的状态转换图(注：该图中 A_1—A_{23}与图 3 的不同仅在于 $A_{12}, A_{16}, A_{21}, A_3$ 中 send WTBYTE 命令(不是 INVALID)，A_3, A_{16}中 copy 值不变)

等概率模型假设图 1 的结构中文件各逻辑块在各处理机的磁盘上等概率分布，即其位于每一结点盘上的概率为 $1/n$；同时各结点逻辑块进行等概率访问，即一 disk cache 块 owner=1 的概率为 $1/n$. 假设每个结点上的精确 LRU 淘汰算法命中率为 d(很高)，与磁盘及总线传输时间相比各处理机的内存操作时间均用微小量 ε 表示. 一次文件读写操作为写的概率为 P_w. 设读或写一盘块的时间为 t_{disk}，总线传输一个命令或一个字节数据的时间为 t_c，传输一块数据时间为 T_c，由于硬件技术的发展，并行计算机系统中的通信时间越来越短. 如在“worm_ hole”技术中，数量级可达到 10^{-6} 秒，故在下面计算中与数量级为 10^{-3} 秒的 t_{disk} 相比，t_c 及 T_c 可忽略. 设各结点上 disk cache 共有 m 块(m 较大，一般在 100n 到 200n 之间). 各方法的平均性能用平均读写时间来衡量，与读相比，写只要多发几条命令，从相对于磁盘操作时间的数量级看，写与读反映的结果是一致的，故此处仅采用一次读操作的平均时间 T 作为性能指标.

1. $$T_{\text{主}} = \frac{1}{n}(d\cdot\varepsilon + (1-d)(T_r + t_{\text{disk}} + \varepsilon)) + \frac{n-1}{n}(d\cdot(1-P_i)\cdot\varepsilon$$
$$+ (1 - d(1-P_i))\cdot(T_r + t_c + (d\cdot\varepsilon + (1-d)(T_r + t_{\text{disk}} + \varepsilon)) + T_c))$$
$$\approx \frac{1}{n}(1-d)(T_r + t_{\text{disk}}) + \frac{n-1}{n}(1 - d + d\cdot P_i)(T_r + (1-d)(T_r + t_{\text{disk}}))$$

其中，P_i 为某一时刻 LB 在其非本地结点上的 disk cache 映象接到 INVALID 命令的概率，在“主从式”方法中 $P_i = P_1 + P_2$，P_1 为由于本地结点上的复本被淘汰而接到 INVALID 命令的概率，P_2 为由于某一复本执行写操作而接到INVALID命令的概率. T_r 为淘汰算法选择一块的平均时间. 设 P_d 为 LB 的本地结点上的 disk cache 映象为 DIRTY 的概率，即至少有一复本进行过写操作的概率；P_c 为除本地结点外其它结点上至少有一复本的概率. 那么：

$$T_r = (1 - \frac{n-1}{n}P_i)^m(\frac{n-1}{n}\cdot t_c + \frac{1}{n}(P_d\cdot t_{\text{disk}} + (1-P_d)\cdot\varepsilon$$
$$+ t_c P_c + (1-P_c)\cdot\varepsilon)) + (1 - (1 - \frac{n-1}{n}P_i)^m)\cdot\varepsilon$$
$$\approx \frac{1}{n}P_d(1 - \frac{n-1}{n}P_i)^m\cdot t_{\text{disk}}$$

(1) 在 WIOC 中，$P_2 = 1 - (1 - \frac{1}{2n}P_w)^{n-1}$；而 m 很大，P_1 很小，故 $P_i \approx P_2, T_r \approx 0$，所以：

$$T_{主}^{WIOC} \approx \frac{1}{n}(1-d)t_{disk} + \frac{n-1}{n}(1-d+dP_2)(1-d)t_{disk}$$

$$= \frac{1}{n}(1-d)(1+(n-1)(1-d+dP_2))t_{disk}$$

(2) 在 WTAC 中，$P_2 = 0$；$P_i = P_1$ 很小，故 $T_r \approx \frac{1}{n}P_d t_{disk}$；而 $P_d = 1-(1-\frac{1}{2n}P_w)^n$，所以：

$$T_{主}^{WTAC} \approx \frac{1}{n}(1-d)(\frac{1}{n}P_d t_{disk} + t_{disk}) + \frac{n-1}{n}(1-d)(\frac{1}{n}P_d t_{disk}$$

$$+ (1-d)(\frac{1}{n}P_d t_{disk} + t_{disk}))$$

$$= \frac{1}{n}(1-d)(\frac{1}{n}P_d + 1 + (n-1)(\frac{1}{n}P_d + (1-d)(\frac{1}{n}P_d + 1)))t_{disk}$$

2. $T_{对} = \frac{1}{n}(d(1-P_i)\varepsilon + (1-d(1-P_i))(T_r + P_c(t_c + T_c) + (1-P_c)t_{disk} + \varepsilon))$

$$+ \frac{n-1}{n}(d(1-P_i)\varepsilon + (1-d(1-P_i))(T_r + t_c + d(1-P_i)\varepsilon$$

$$+ (1-d(1-P_i))(P_c' t_c + (1-P_c')(T_r + t_{disk} + \varepsilon)) + T_c))$$

$$\approx \frac{1}{n}(1-d+dP_i)(T_r + (1-P_c)t_{disk}) + \frac{n-1}{n}(1-d+dP_i)$$

$$(T_r + (1-d+dP_i)(1-P_c')(T_r + t_{disk}))$$

其中，P_i，P_c 及 T_r 含义同上，P_c' 为 LB 除去在某一非本地结点及本地结点上的 disk cache 映象外，其它结点上至少有一复本的概率，故：

$$P_c = 1-(1-\frac{1}{2n})^{n-1};\ P_c' = 1-(1-\frac{1}{2n})^{n-2}$$

设 P_c'' 为LB除去在某一结点上的disk cache映象外，其它结点上至少有一复本的概率，故 $P_c'' = P_c$. 又设 P_d 为 LB 的任一 disk cache 映象状态为 DIRTY 的概率，那么：

$$T_r = (1-\frac{n-1}{n}P_i)^m(\frac{n-1}{n}(P_d(t_c + P_c'' t_c + (1-P_c''))(T_c + t_{disk}) + t_c + t_c)$$

$$+ (1-P_d)t_c) + \frac{1}{n}(P_d(P_c'' t_c + (1-P_c'')t_{disk}) + (1-P_d)\varepsilon)) + (1-(1-\frac{n-1}{n}P_i)^m \cdot \varepsilon$$

$$\approx (1-\frac{n-1}{n}P_i)^m(\frac{n-1}{n}P_d(1-P_c'')t_{disk} + \frac{1}{n}P_d(1-P_c'')t_{disk})$$

$$= P_d(1-P_c'')(1-\frac{n-1}{n}P_i)^m t_{disk}$$

(1) 在 WIOC 中，$P_i = P_2 = 1-(1-\frac{1}{2n}P_w)^{n-1}$，$T_r \approx 0$，所以：

$$T_{对}^{WIOC} \approx \frac{1}{n}(1-d+dP_i)(1-P_c + (n-1)(1-d+dP_i)(1-P_c'))t_{disk}$$

(2) 在 WTAC 中，$P_i = 0$，$T_r = P_d(1-P_c'')t_{disk} = P_d(1-P_c)t_{disk}$；$P_d \approx P_w(1-\frac{1}{2n}P_w)^{n-1}$，所以：

$$T_{对}^{WTAC} \approx \frac{1}{n}(1-d)(P_d(1-P_c) + (1-P_c) + (n-1)(P_d(1-P_c)$$

$$+ (1-d)(1-P_c')(P_d(1-P_c) + 1)))t_{disk}$$

为与 disk cache 单复本系统相比较，对单复本系统的一次读操作平均时间 $T_{单}$ 估算如下：

$$3.\ T_{单}=\frac{1}{n}(d\cdot\varepsilon+(1-d)(T_r+t_{disk}+\varepsilon))+\frac{n-1}{n}(t_c+(d\cdot\varepsilon+(1-d)(T_r+t_{disk}+\varepsilon)+T_c))$$
$$\approx(1-d)(T_r+t_{disk});$$

其中，T_r 含义同上，设 P_d 为一 disk cache 块状态为 DIRTY 的概率，则：

$$T_r=P_d\cdot t_{disk}+(1-P_d)\cdot\varepsilon\approx P_d\cdot t_{disk};\quad P_d=1-(1-\frac{1}{2n}P_w)^n$$

在实际应用中，文件的只读情况占很大比例，此时 $P_w=0$，取 $d=90\%$，$n=4$—10 以及 $n=8$，$d=70\%$—70.5%，计算上述各多复本方法和单复本方法的性能分析结果分别如表 2(a)，(b) 所示；对于一般情况，读操作比写操作的频度大得多，取 $P_w=30\%$，其它各参数同上，计算此时各性能分析结果如表 3(a)，(b) 所示.

表 2(a)　($P_w=0$(只读)，$d=90\%$，单位：t_{disk})

n	4	5	6	7	8	9	10
单副本	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
主从式	0.0325	0.0280	0.0250	0.0229	0.0213	0.0200	0.0190
对称式	0.0225	0.0190	0.0167	0.0151	0.0139	0.0130	0.0123

表 2(b)　($P_w=0$(只读)，$n=8$，单位：t_{disk})

d	70%	72.5%	75%	77.5%	80%	82.5%	85%	87.5%	90%	92.5%	95%	97.5%
单副本	0.3000	0.2750	0.2500	0.2250	0.2000	0.1750	0.1500	0.1250	0.1000	0.0750	0.0500	0.0250
主从式	0.1163	0.1005	0.0859	0.0724	0.0600	0.0487	0.0384	0.0293	0.0213	0.0143	0.0084	0.0037
对称式	0.0773	0.0668	0.0570	0.0480	0.0397	0.0321	0.0253	0.0192	0.0139	0.0093	0.0055	0.0024

表 3(a)　($P_w=30\%$，$d=90\%$，单位：t_{disk})

n		4	5	6	7	8	9	10
单副本		0.1142	0.1141	0.1141	0.1141	0.1140	0.1140	0.1140
主从式	WIOC	0.0398	0.0363	0.0339	0.0323	0.0310	0.0300	0.0293
	WTAC	0.0354	0.0305	0.0271	0.0247	0.0229	0.0215	0.0204
对称式	WIOC	0.0555	0.0508	0.0475	0.0452	0.0435	0.0421	0.0410
	WTAC	0.0414	0.0374	0.0348	0.0330	0.0316	0.0306	0.0298

表 3(b)　($P_w=30\%$，$n=8$，单位：t_{disk})

d		70%	72.5%	75%	77.5%	80%	82.5%	85%	87.5%	90%	92.5%	95%	97.5%
单副本		0.3421	0.3136	0.2851	0.2566	0.2281	0.1996	0.1711	0.1426	0.1141	0.0855	0.0570	0.0285
主从式	WIOC	0.1391	0.1222	0.1063	0.0914	0.0774	0.0643	0.0523	0.0412	0.0310	0.0218	0.0136	0.0063
	WTAC	0.1221	0.1058	0.0907	0.0766	0.0636	0.0518	0.0411	0.0314	0.0229	0.0155	0.0092	0.0041
对称式	WIOC	0.1197	0.1082	0.0972	0.0868	0.0770	0.0678	0.0591	0.0510	0.0435	0.0365	0.0301	0.0243
	WTAC	0.1365	0.1203	0.1050	0.0906	0.0771	0.0644	0.0526	0.0417	0.0316	0.0224	0.0141	0.0066

5 结束语

由以上性能分析的结果可知，从并行计算机应用的通用性考虑，在并行文件系统中引入 disk cache 多复本对文件系统性能有显著提高，与单复本策略相比，其平均性能的提高均接近一个数量级. 从平均性能的角度来看，在只读应用系统中，对称式维护方法比主从式要好些；对于一般的应用系统，在命中率较高时，主从式方法比对称式稍好些；各类方法中 WTAC 要比 WIOC 好.

对于上述各多复本方法及单复本方法，已在 BJ-1 并行计算机上进行了模拟，限于篇幅，仅将该模拟结果的部分数据列于表 4(a)，(b) 中，该模拟测试的结果与上述理论分析结果基本相同. 今后将对此深入研究，针对实际情况对各方法进行改进.

表 4(a) （读取逻辑文件块数：3000，$P_w = 0$(只读)，$d = 87.2\%$，$n = 4$）

	实际读取盘块次数	通信次数	一次读操作的平均读盘时间	一次读操作的平均通信次数
单副本	383	4578	0.128 t_{disk}	1.526
主从式	142	1084	0.047 t_{disk}	0.361
对称式	110	1092	0.037 t_{disk}	0.364

表 4(b) （读写逻辑文件块数：1400，$P_w = 30\%$，$d = 68\%$，$n = 4$）

		实际读写盘块次数	通信次数	一次读写操作的平均盘操作时间	一次读写操作的平均通信次数
单副本		658	2978	0.470 t_{disk}	2.13
主从式	WIOC	243	2877	0.174 t_{disk}	2.055
	WTAC	238	2394	0.170 t_{disk}	1.710
对称式	WIOC	230	3366	0.164 t_{disk}	2.404
	WTAC	251	2803	0.179 t_{disk}	2.002

参 考 文 献

[1] Accetta M *et al*. Mach: a new kernel foundation for UNIX development. In: Proc Summer 1986 USENIX Conf, 1986, 93－112.

[2] Mullender S J *et al*. Amoeba: a distributed operating system for 1990s. *IEEE Computer*, 1990, 23(5): 44－53.

[3] French J C *et al*. Performance measurement of a parallel input/output system for the Intel ipsc /2 hypercube. *ACM Performance Evaluation Review*, 1991, 18(4): 178－187.

[4] 武北虹等. 并行文件系统 PFS 的设计与分析. 软件学报, 1995, 6(11): 641－646.

[5] 尤晋元. UNIX 操作系统教程. 西安：西北电讯工程学院出版社, 1985.

[6] Kotz D, Ellis C S. Caching and writeback policies in parallel file system. *Journal of Parallel and Distributed Computing*, 1993, 17(1—2): 140－145.

[7] Archibald J, Baer J L. Cache coherence protocols: evaluation using a multiprocessor simulation model. *ACM Trans on Computer System*, 1986, 4(4): 273－298.

[8] Eggers S, Katz R. Evaluating the performance of four snooping cache coherency protocols. In: Proc 16th Int Symp Computer Architecture, ACM, 1989, 2－15.

[9] Lenoski D *et al*. The directory-based cache coherence protocol for the DASH multiprocessor. In: Proc 17th Int Symp Computer Architecture, IEEE, 1990, 148－159.